

Лабораторная работа №6

Мандатное разграничение прав в Linux.

Галиева Аделина Руслановна

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выводы	15

Список иллюстраций

2.1	Запуск httpd	7
2.2	Запуск httpd	8
2.3	Контекст безопасности	9
2.4	Переключатели SELinux для httpd	10
2.5	Создание html-файла и доступ по httpd	11
2.6	Ошибка доступа после изменения контекста	12
2.7	Лог ошибок	12
2.8	Переключение порта	13
2.9	Доступ по http на 81 порт	14

Список таблиц

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 Выполнение лабораторной работы

1. Устанавливаем `httpd`.
2. Задаем имя сервера.
3. Открываем порты для работы с протоколом `httpd`.

```
argalieva@username:/var/www/html

grub2-tools-efi-1:2.06-70.el9_3.2.rocky.0.5.x86_64
grub2-tools-extra-1:2.06-70.el9_3.2.rocky.0.5.x86_64
kernel-5.14.0-362.24.1.el9_3.0.1.x86_64
kernel-core-5.14.0-362.24.1.el9_3.0.1.x86_64
kernel-modules-5.14.0-362.24.1.el9_3.0.1.x86_64
kernel-modules-core-5.14.0-362.24.1.el9_3.0.1.x86_64

Выполнено!
[root@username argalieva]# dnf install httpd
Последняя проверка окончания срока действия метаданных: 0:31:00 назад, Ср 24 а
пр 2024 12:34:27.
Зависимости разрешены.
=====
Пакет                Архитектура          Версия                Репозиторий          Размер
=====
Установка:
  httpd                x86_64                2.4.57-5.el9          appstream              46 k
Установка зависимостей:
  apr                  x86_64                1.7.0-12.el9_3        appstream              122 k
  apr-util              x86_64                1.6.1-23.el9          appstream              94 k
  apr-util-bdb          x86_64                1.6.1-23.el9          appstream              12 k
  httpd-core            x86_64                2.4.57-5.el9          appstream              1.4 M
  httpd-filesystem      noarch                2.4.57-5.el9          appstream              13 k
  httpd-tools           x86_64                2.4.57-5.el9          appstream              80 k
  rocky-logos-httpd     noarch                90.15-2.el9           appstream              24 k
Установка слабых зависимостей:
  apr-util-openssl      x86_64                1.6.1-23.el9          appstream              14 k
  mod_http2             x86_64                1.15.19-5.el9_3.1     appstream              148 k
  mod_lua               x86_64                2.4.57-5.el9          appstream              60 k

Результат транзакции
=====
Установка 11 Пакетов

Объем загрузки: 2.0 М
Объем изменений: 6.0 М
Продолжить? [д/н]: д
Загрузка пакетов:
(1/11): mod_lua-2.4.57-5.el9.x86_64.rpm      49 kB/s | 60 kB      00:01
(2/11): rocky-logos-httpd-90.15-2.el9.noarch. 19 kB/s | 24 kB      00:01
(3/11): httpd-tools-2.4.57-5.el9.x86_64.rpm  62 kB/s | 80 kB      00:01
```

Рис. 2.1: Запуск httpd

```
argalieva@username:/var/www/html

apr-util-bdb-1.6.1-23.el9.x86_64      apr-util-openssl-1.6.1-23.el9.x86_64
httpd-2.4.57-5.el9.x86_64            httpd-core-2.4.57-5.el9.x86_64
httpd-filesystem-2.4.57-5.el9.noarch  httpd-tools-2.4.57-5.el9.x86_64
mod_http2-1.15.19-5.el9_3.1.x86_64   mod_lua-2.4.57-5.el9.x86_64
rocky-logos-httpd-90.15-2.el9.noarch

Выполнено!
[root@username argalieva]# service httpd start
Redirecting to /bin/systemctl start httpd.service
[root@username argalieva]# 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:
   Active: active (running) since Wed 2024-04-24 13:07:02 MSK; 14s ago
     Docs: man:httpd.service(8)
    Main PID: 94948 (httpd)
   Status: "Total requests: 0; Idle/Busy workers 100/0;Requests/sec: 0; Byt
     Tasks: 213 (limit: 10888)
    Memory: 27.5M
       CPU: 286ms
    CGroup: /system.slice/httpd.service
            └─94948 /usr/sbin/httpd -DFOREGROUND
              └─94949 /usr/sbin/httpd -DFOREGROUND
                └─94950 /usr/sbin/httpd -DFOREGROUND
                  └─94951 /usr/sbin/httpd -DFOREGROUND
                    └─94952 /usr/sbin/httpd -DFOREGROUND

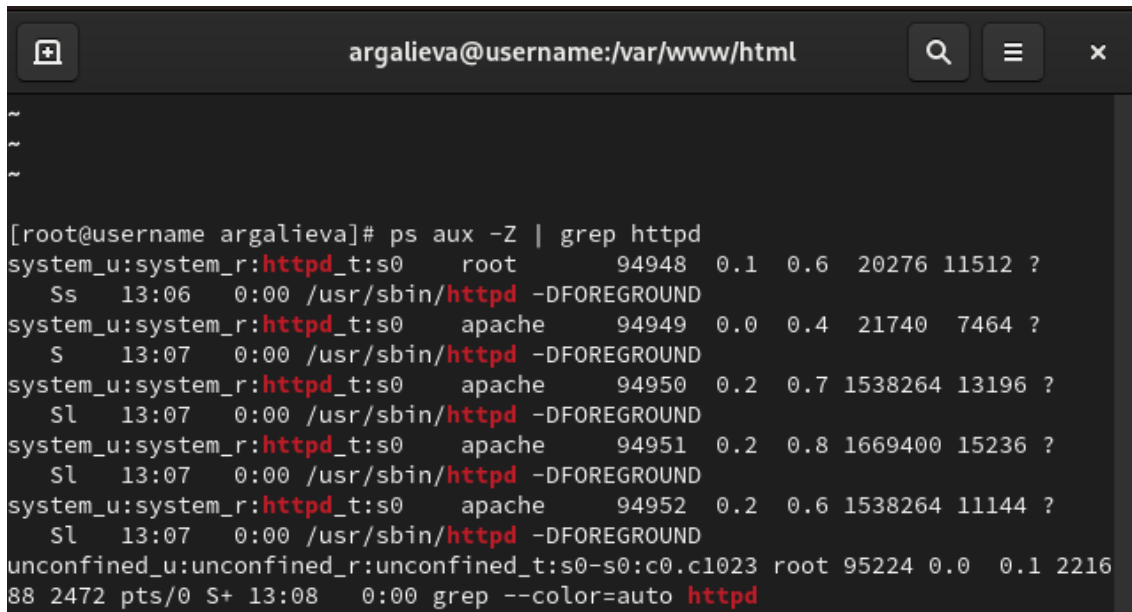
anp 24 13:06:26 username systemd[1]: Starting The Apache HTTP Server...
anp 24 13:06:46 username httpd[94948]: AH00558: httpd: Could not reliably det
anp 24 13:07:02 username systemd[1]: Started The Apache HTTP Server.
anp 24 13:07:02 username httpd[94948]: Server configured, listening on: port >
lines 1-20/20 (END)...skipping...
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset:
   Active: active (running) since Wed 2024-04-24 13:07:02 MSK; 14s ago
     Docs: man:httpd.service(8)
    Main PID: 94948 (httpd)
   Status: "Total requests: 0; Idle/Busy workers 100/0;Requests/sec: 0; Byt
     Tasks: 213 (limit: 10888)
    Memory: 27.5M
       CPU: 286ms
    CGroup: /system.slice/httpd.service
```

Рис. 2.2: Запуск httpd

4. Входим в систему с полученными учётными данными и убеждаемся, что SELinux работает в режиме enforcing политики targeted с помощью команд `getenforce` и `sestatus`.
5. Обращаемся с помощью браузера к веб-серверу, запущенному на вашем

компьютере, и убеждаемся, что последний работает: `service httpd status` или `/etc/rc.d/init.d/httpd status`.

6. Находим веб-сервер Apache в списке процессов, определяем его контекст безопасности и заносим эту информацию в отчёт. Используем команду `ps auxZ | grep httpd`.



```
argalieva@username:/var/www/html

~
~
~

[root@username argalieva]# ps aux -Z | grep httpd
system_u:system_r:httpd_t:s0 root 94948 0.1 0.6 20276 11512 ?
Ss 13:06 0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 94949 0.0 0.4 21740 7464 ?
S 13:07 0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 94950 0.2 0.7 1538264 13196 ?
Sl 13:07 0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 94951 0.2 0.8 1669400 15236 ?
Sl 13:07 0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 94952 0.2 0.6 1538264 11144 ?
Sl 13:07 0:00 /usr/sbin/httpd -DFOREGROUND
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 root 95224 0.0 0.1 2216
88 2472 pts/0 S+ 13:08 0:00 grep --color=auto httpd
```

Рис. 2.3: Контекст безопасности

7. Смотрим текущее состояние переключателей SELinux для Apache с помощью команды `sestatus -bigrep httpd`. Обращаем внимание, что многие из них находятся в положении «off»

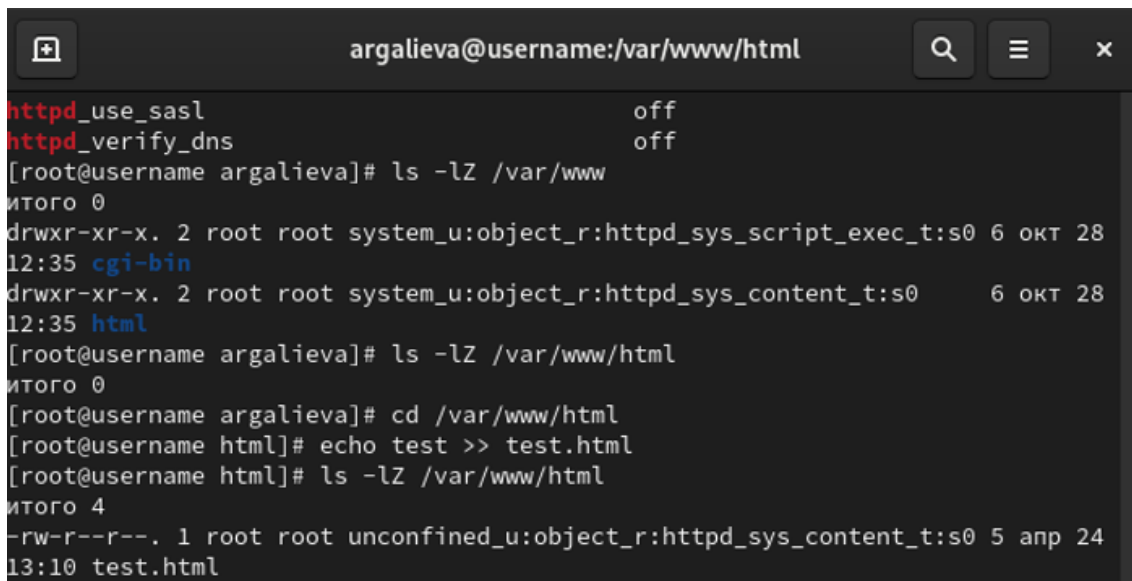
```
[root@username argalieva]# sestatus -b | grep httpd
httpd_anon_write off
httpd_builtin_scripting on
httpd_can_check_spam off
httpd_can_connect_ftp off
httpd_can_connect_ldap off
httpd_can_connect_mythtv off
httpd_can_connect_zabbix off
httpd_can_manage_courier_spool off
httpd_can_network_connect 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_avaahi 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
```

Рис. 2.4: Переключатели SELinux для httpd

8. Смотрим статистику по политике с помощью команды `seinfo`, также определяем множество пользователей, ролей, типов.
9. Определяем тип файлов и поддиректорий, находящихся в директории `/var/www`, с помощью команды `ls -lZ /var/www`. В поддиректориях могут располагаться системные скрипты и контент для httpd.
10. Определяем тип файлов, находящихся в директории `/var/www/html`: `ls -lZ /var/www/html`. В директории изначально нет файлов.
11. Определяем круг пользователей, которым разрешено создание файлов в директории `/var/www/html`. Создаем файлы может только root.
12. Создаем от имени суперпользователя (так как в дистрибутиве после установки только ему разрешена запись директорию) html-файл `/var/www/html/test.html` следующего содержания:

test

13. Проверяем контекст созданного вами файла. Заносим в отчёт контекст, присваиваемый по умолчанию вновь созданным файлам в директории /var/www/html.
14. Обращаемся к файлу через веб-сервер, введя в браузере адрес `http://127.0.0.1/test.html`. Убеждаемся, что файл был успешно отображён.



```
argalieva@username:/var/www/html
httpd_use_sasl off
httpd_verify_dns off
[root@username argalieva]# ls -lZ /var/www
итого 0
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_script_exec_t:s0 6 окт 28
12:35 cgi-bin
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_content_t:s0 6 окт 28
12:35 html
[root@username argalieva]# ls -lZ /var/www/html
итого 0
[root@username argalieva]# cd /var/www/html
[root@username html]# echo test >> test.html
[root@username html]# ls -lZ /var/www/html
итого 4
-rw-r--r--. 1 root root unconfined_u:object_r:httpd_sys_content_t:s0 5 апр 24
13:10 test.html
```

Рис. 2.5: Создание html-файла и доступ по httpd

15. Изучаем справку `man httpd_selinux` и выясняем, какие контексты файлов определены для httpd. Сопоставляем их с типом файла `test.html`. Проверяем контекст файла можно командой `ls -Z`. `ls -Z /var/www/html/test.html`. Основным контекстом является `httpd_sys_content_t`, его мы и увидим в выводе команды.
16. Изменяем 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`: `chcon -t samba_share_t /var/www/html/test.html` `ls -Z /var/www/html/test.html`. После этого проверяем, что контекст поменялся.

17. Пробуем ещё раз получить доступ к файлу через веб-сервер, введя в браузере адрес `http://127.0.0.1/test.html`. Мы должны получить сообщение об ошибке: `Forbidden You don't have permission to access /test.html on this server`. При изменении контекста файл стал считаться чужим для `http` и программа не может его прочитать.

```
[root@username html]# chcon -t samba_share_t test.html
[root@username html]# ls -lZ /var/www/html
итого 4
-rw-r--r--. 1 root root unconfined_u:object_r:samba_share_t:s0 5 апр 24 13:10
test.html
```

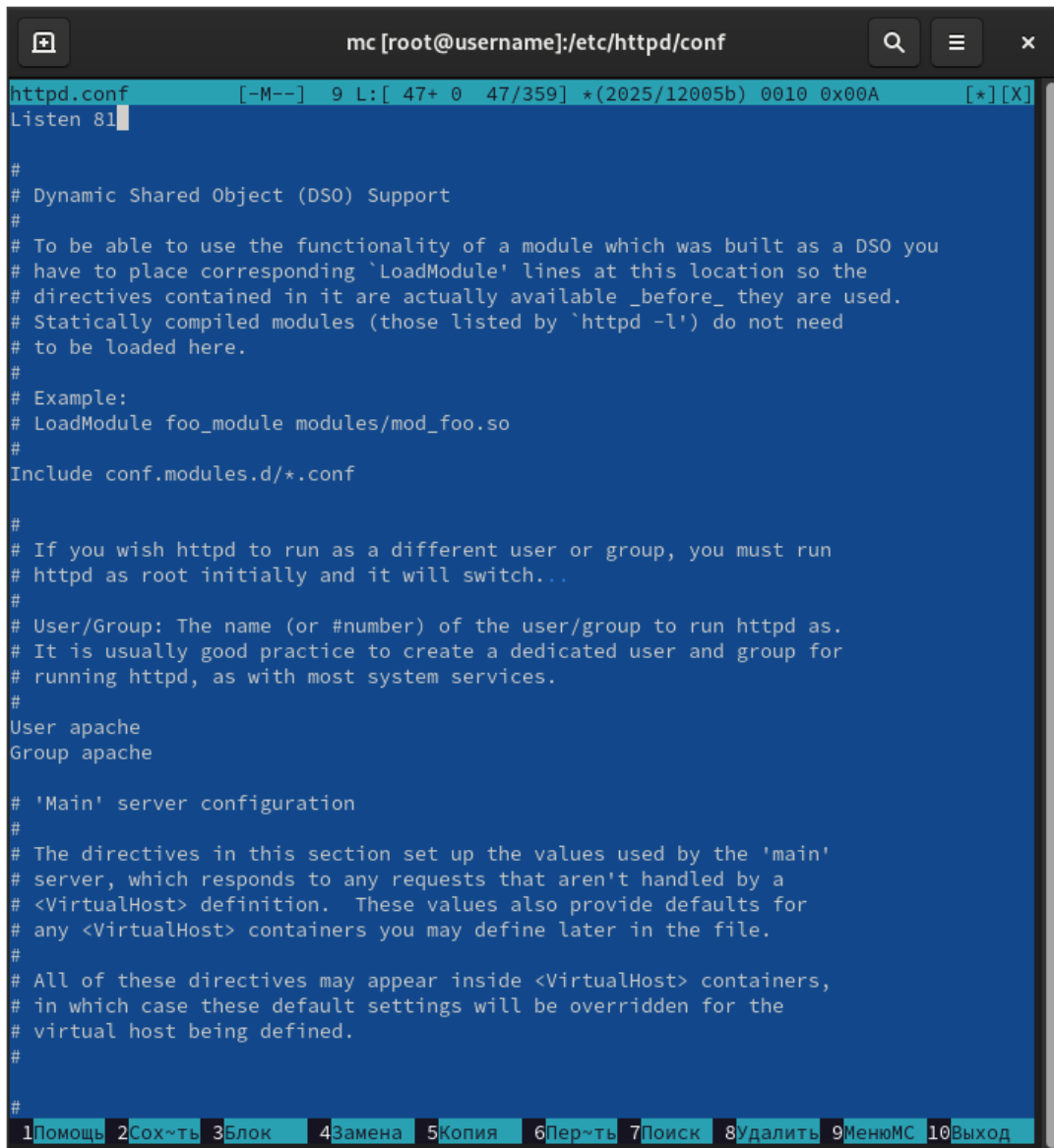
Рис. 2.6: Ошибка доступа после изменения контекста

18. Проанализируем ситуацию. Почему файл не был отображён, если права доступа позволяют читать этот файл любому пользователю? `ls -l /var/www/html/test.html`. Просматриваем log-файлы веб-сервера Apache. Также просматриваем системный лог-файл: `tail /var/log/messages`. Если в системе окажутся запущенными процессы `setroubleshootd` и `audtd`, то вы также сможете увидеть ошибки, аналогичные указанным выше, в файле `/var/log/audit/audit.log`. Проверяем это утверждение самостоятельно.

```
[root@username html]# tail /var/log/messages
Apr 24 13:06:59 username kernel: e1000: enp0s3 NIC Link is Down
Apr 24 13:07:02 username systemd[1]: Started The Apache HTTP Server.
Apr 24 13:07:02 username httpd[94948]: Server configured, listening on: port 80
Apr 24 13:07:05 username kernel: e1000: enp0s3 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: RX
Apr 24 13:07:05 username NetworkManager[892]: <info> [1713953225.6485] device (enp0s3): carrier: link connected
Apr 24 13:07:47 username chronyd[737]: Selected source 95.31.7.160 (2.rocky.pool.ntp.org)
Apr 24 13:08:25 username systemd[1]: Starting Fingerprint Authentication Daemon...
Apr 24 13:08:26 username systemd[1]: Started Fingerprint Authentication Daemon.
Apr 24 13:08:29 username NetworkManager[892]: <info> [1713953309.5432] agent-manager: agent[08d8fdb170ee32c,:1.69/org.gnome.Shell.NetworkAgent/1000]: agent registered
Apr 24 13:08:56 username systemd[1]: fprintd.service: Deactivated successfully
```

Рис. 2.7: Лог ошибок

19. Пробуем запустить веб-сервер Apache на прослушивание TCP-порта 81 (а не 80, как рекомендует IANA и прописано в /etc/services). Для этого в файле /etc/httpd/httpd.conf находим строчку Listen 80 и заменяем её на Listen 81.



The image shows a text editor window titled 'mc [root@username]:/etc/httpd/conf'. The file being edited is 'httpd.conf'. The current line is 'Listen 81', which has been changed from the default 'Listen 80'. The editor's status bar at the top shows '9 L: [47+ 0 47/359] *(2025/12005b) 0010 0x00A [*] [X]'. The bottom of the window features a menu bar with the following items: 1Помощь, 2Сохранить, 3Блок, 4Замена, 5Копия, 6Перейти, 7Поиск, 8Удалить, 9МенюМС, 10Выход.

```
httpd.conf [-M--] 9 L: [ 47+ 0 47/359] *(2025/12005b) 0010 0x00A [*] [X]
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

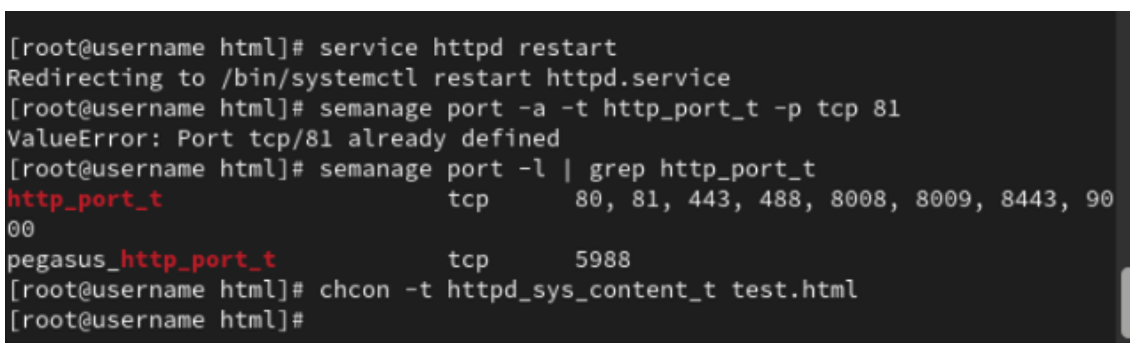
#
# If you wish httpd to run as a different user or group, you must run
# httpd as root initially and it will switch...
#
# User/Group: The name (or #number) of the user/group to run httpd as.
# It is usually good practice to create a dedicated user and group for
# running httpd, as with most system services.
#
User apache
Group apache

# 'Main' server configuration
#
# The directives in this section set up the values used by the 'main'
# server, which responds to any requests that aren't handled by a
# <VirtualHost> definition. These values also provide defaults for
# any <VirtualHost> containers you may define later in the file.
#
# All of these directives may appear inside <VirtualHost> containers,
# in which case these default settings will be overridden for the
# virtual host being defined.
#
#
```

Рис. 2.8: Переключение порта

20. Выполняем перезапуск веб-сервера Apache. Произошёл сбой? Сбой не происходит, порт 81 уже вписан в разрешенные.

21. Проанализируем лог-файлы: `tail -nl /var/log/messages`. Просматриваем файлы `/var/log/http/error_log`, `/var/log/http/access_log` и `/var/log/audit/audit.log` и выясняем, в каких файлах появились записи.
22. Выполняем команду `semanage port -a -t http_port_t -p tcp 81`. После этого проверяем список портов командой `semanage port -l | grep http_port_t`. Убеждаемся, что порт 81 появился в списке.
23. Пробуем запустить веб-сервер Apache ещё раз.
24. Возвращаем контекст `httpd_sys_content_t` к файлу `/var/www/html/test.html`: `chcon -t httpd_sys_content_t /var/www/html/test.html`. После этого пробуем получить доступ к файлу через веб-сервер, введя в браузере адрес `http://127.0.0.1:81/test.html`. Мы видим содержимое файла — слово «test».



```
[root@username html]# service httpd restart
Redirecting to /bin/systemctl restart httpd.service
[root@username html]# semanage port -a -t http_port_t -p tcp 81
ValueError: Port tcp/81 already defined
[root@username html]# 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@username html]# chcon -t httpd_sys_content_t test.html
[root@username html]#
```

Рис. 2.9: Доступ по http на 81 порт

25. Исправляем обратно конфигурационный файл `apache`, вернув `Listen 80`.
26. Удаляем привязку `http_port_t` к 81 порту: `semanage port -d -t http_port_t -p tcp 81` и проверяем, что порт 81 удалён.
27. Удаляем файл `/var/www/html/test.html`: `rm /var/www/html/test.html`.

3 Выводы

В ходе выполнения лабораторной работы я развила навыки администрирования ОС Linux. Получила первое практическое знакомство с технологией SELinux. Проверила работу SELinux на практике совместно с веб-сервером Apache.