

Отчёт по лабораторной работе №6

Дисциплина: Основы информационной безопасности

Балакирева Дарья Сергеевна, НПМбд-01-196

Содержание

1	Цель работы	5
2	Теоретическое введение	6
3	Выполнение лабораторной работы	8
4	Выводы	15
	Список литературы	16

Список иллюстраций

3.1	getenforce и sestatus	8
3.2	Работающий сервер	9
3.3	Контекст безопасности Apache	9
3.4	Состояние переключателей	9
3.5	Статистика seinfo	10
3.6	Данные директорий /var/www и /var/www/html	10
3.7	Файл test.html	10
3.8	Контекст файла test.html	11
3.9	Просмотр файла в веб-браузере	11
3.10	Смена контекста	11
3.11	Ошибка доступа	12
3.12	Ошибки в log-файлах	12
3.13	Прослушивание 81 порта	13
3.14	Перезапуск сервера	13
3.15	Установка порта	13
3.16	Повторный просмотр файла в веб-браузере	14
3.17	Удаление порта	14
3.18	Удаление файла	14

Список таблиц

1 Цель работы

Получить практические навыки администрирования в ОС Linux и ознакомиться с технологией SELinux совместно с веб-сервером Apache.

2 Теоретическое введение

SELinux, или Security Enhanced Linux, — это продвинутый механизм управления доступом, разработанный Агентством национальной безопасности (АНБ) США для предотвращения злонамеренных вторжений. Он реализует мандатную модель управления доступом (MAC — Mandatory Access control) в дополнение к уже существующей в Linux дискреционной модели (DAC — Discretionary Access Control), то есть разрешениям на чтение, запись, выполнение.

У SELinux есть три режима работы:

- Enforcing — ограничение доступа в соответствии с политикой. Запрещено все, что не разрешено в явном виде. Режим по умолчанию.
- Permissive — ведёт лог действий, нарушающих политику, которые в режиме enforcing были бы запрещены, но не запрещает сами действия.
- Disabled — полное отключение SELinux.

В основе структуры безопасности SELinux лежат политики. Политика — это набор правил, определяющих ограничения и права доступа для всего, что есть в системе. Под “всем” в данном случае понимаются пользователи, роли, процессы и файлы. Политика определяет связь этих категорий друг с другом. |

Более подробно см. в [1].

Apache — это свободное программное обеспечение, с помощью которого можно создать веб-сервер. Несмотря на то, что Apache чаще всего называют сервером (более того, его официальное название — Apache HTTP Server) — это всё-таки программа, которую устанавливают на сервер, чтобы добиться определённых результатов.

Для чего нужен Apache сервер:

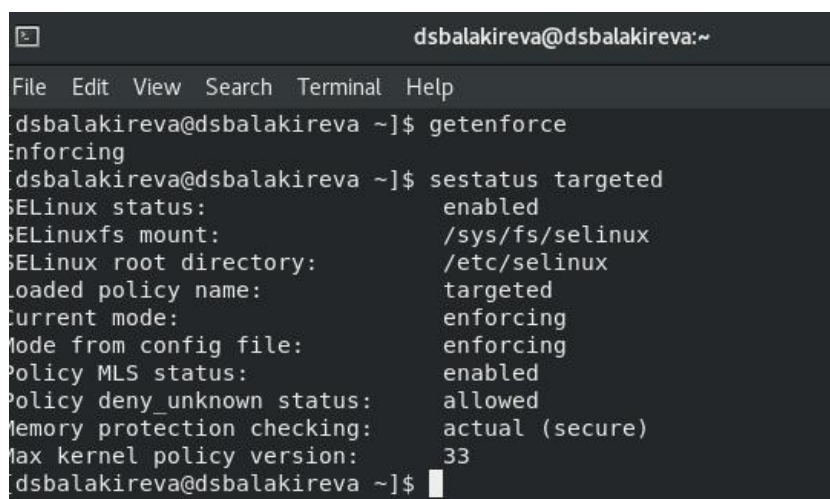
- чтобы открывать динамические PHP-страницы,
- для распределения поступающей на сервер нагрузки,
- для обеспечения отказоустойчивости сервера,
- чтобы потренироваться в настройке сервера и запуске PHP-скриптов.

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

Более подробно см. в **[gnu-doc-1:bash?]**.

3 Выполнение лабораторной работы

С помощью команды `getenforce` убеждаемся, что SELinux работает в режиме `enforcing`, а с помощью команды `sestatus` устанавливаем политику `targeted` (рис. 3.1).



```
dsbalakireva@dsbalakireva:~$ getenforce
enforcing
dsbalakireva@dsbalakireva:~$ sestatus targeted
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
dsbalakireva@dsbalakireva:~$
```

Рис. 3.1: `getenforce` и `sestatus`

Убеждаемся, что сервер работает с помощью команды `service httpd status` (рис. 3.2).


```
[dsbalakireva@dsbalakireva ~]$ sudo systemctl start httpd
[dsbalakireva@dsbalakireva ~]$ 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; vendor prese
   Active: active (running) since Thu 2022-10-13 16:05:30 MSK; 2min 5s ago
     Docs: man:httpd.service(8)
   Main PID: 37708 (httpd)
    Status: "Running, listening on: port 80"
      Tasks: 213 (limit: 12246)
     Memory: 26.4M
    CGroup: /system.slice/httpd.service
            └─37708 /usr/sbin/httpd -DFOREGROUND
              └─37715 /usr/sbin/httpd -DFOREGROUND
                └─37716 /usr/sbin/httpd -DFOREGROUND
                  └─37717 /usr/sbin/httpd -DFOREGROUND
                    └─37718 /usr/sbin/httpd -DFOREGROUND

Oct 13 16:05:30 dsbalakireva.localdomain systemd[1]: Starting The Apache HTTP S
Oct 13 16:05:30 dsbalakireva.localdomain systemd[1]: Started The Apache HTTP Se
Oct 13 16:05:30 dsbalakireva.localdomain httpd[37708]: Server configured, liste
```

Рис. 3.2: Работающий сервер

С помощью команды `ps -eZ` находим, что контекст безопасности Apache — `httpd_t` (рис. 3.3).

```
[dsbalakireva@dsbalakireva ~]$ ps -eZ | grep httpd
system_u:system_r:httpd_t:s0      37708 ?        00:00:00 httpd
system_u:system_r:httpd_t:s0      37715 ?        00:00:00 httpd
system_u:system_r:httpd_t:s0      37716 ?        00:00:00 httpd
system_u:system_r:httpd_t:s0      37717 ?        00:00:00 httpd
system_u:system_r:httpd_t:s0      37718 ?        00:00:00 httpd
```

Рис. 3.3: Контекст безопасности Apache

Смотрим текущее состояние переключателей командой `sestatus -b httpd` (рис. 3.4).

```
[dsbalakireva@dsbalakireva ~]$ 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_yubikey                off
awstats_purge_apache_log_files   off
boinc_execmem                    on
cdrecord_read_content            off
cluster_can_network_connect      off
cluster_manage_all_files         off
```

Рис. 3.4: Состояние переключателей

Смотрим статистику по политике командой `seinfo`. Узнаём, что множество пользователей — 8, ролей — 14, типов — 4982 (рис. 3.5).

```
[dsbalakireva@dsbalakireva ~]$ seinfo
Statistics for policy file: /sys/fs/selinux/policy
Policy Version:          31 (MLS enabled)
Target Policy:           selinux
Handle unknown classes:  allow
Classes:                 132      Permissions:          464
Sensitivities:           1        Categories:           1024
Types:                   4982     Attributes:            255
Users:                   8         Roles:                14
Booleans:                339     Cond. Expr.:          387
Allow:                   112932   Neverallow:           0
Auditallow:              166     Dontaudit:             10378
Type_trans:              252848   Type_change:           87
Type_member:              35     Range_trans:           5782
Role_allow:              38     Role_trans:            421
Constraints:             72     Validatetrans:         0
MLS Constrain:           72     MLS Val. Tran:         0
Permissives:             0       Polcap:                5
Defaults:                7       Typebounds:            0
Allowxperm:              0       Neverallowxperm:       0
Auditallowxperm:         0       Dontauditxperm:        0
Ibendportcon:            0       Ibpkeycon:             0
Initial SIDs:            27       Fs_use:                34
Genfscon:                107     Portcon:               646
Netifcon:                0       Nodecon:               0
```

Рис. 3.5: Статистика seinfo

Определяем тип файлов и круг пользователей с правой на создание и поддиректорий в директориях /var/www и /var/www/html командой `ls -lZ` (рис. 3.6).

```
[dsbalakireva@dsbalakireva ~]$ ls -lZ /var/www
total 0
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_script_exec_t:s0 6 Jun 22 17
:18 cgi-bin
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_content_t:s0 6 Jun 22 17
:18 html
[dsbalakireva@dsbalakireva ~]$ ls -lZ /var/www/html
total 0
[dsbalakireva@dsbalakireva ~]$ ls -l /var/www/html
total 0
[dsbalakireva@dsbalakireva ~]$ ls -l /var/www
total 0
drwxr-xr-x. 2 root root 6 Jun 22 17:18 cgi-bin
drwxr-xr-x. 2 root root 6 Jun 22 17:18 html
[dsbalakireva@dsbalakireva ~]$
```

Рис. 3.6: Данные директорий /var/www и /var/www/html

От имени суперпользователя создаём файл /var/www/html/test.html (рис. 3.7).

```
[dsbalakireva@dsbalakireva ~]$ su -
Password:
[root@dsbalakireva ~]# touch /var/www/html/test.html
[root@dsbalakireva ~]# nano /var/www/html/test.html
[root@dsbalakireva ~]# cat /var/www/html/test.html
<html>
<body>test</body>
</html>
```

Рис. 3.7: Файл test.html

Командой `matchpathcon` узнаём контекст файла `test.html` и директории `/var/www/html` — это `httpd_sys_content_t` (рис. [-#fig:008]).

```
[root@dsbalakireva ~]# matchpathcon /var/www/html/test.html
/var/www/html/test.html system_u:object_r:httpd_sys_content_t:s0
[root@dsbalakireva ~]# matchpathcon -V /var/www/html
/var/www/html verified.
[root@dsbalakireva ~]# matchpathcon /var/www/html
/var/www/html system_u:object_r:httpd_sys_content_t:s0
[root@dsbalakireva ~]#
```

Рис. 3.8: Контекст файла `test.html`

Обращаемся к файлу через ссылку в веб-браузере. Контент отображён корректно (рис. 3.9).

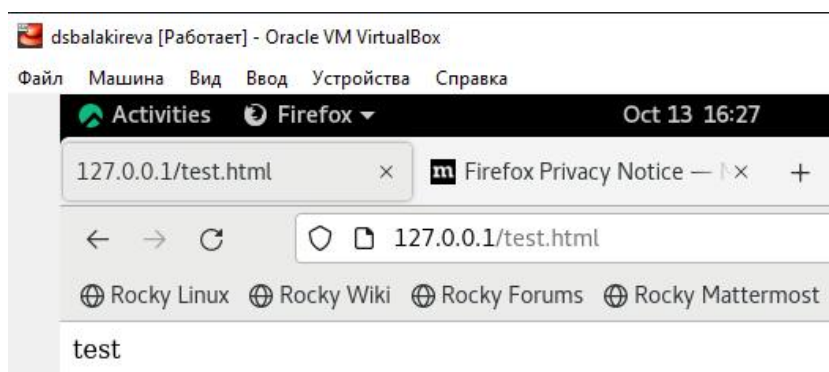


Рис. 3.9: Просмотр файла в веб-браузере

Изучая справку `man httpd_selunix` узнаём, что для `httpd` определены следующие контексты: `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`. Меняем контекст файла `test.html` командой `chcon -t` (рис. 3.10).

```
[root@dsbalakireva ~]# ls -Z /var/www/html/test.html
unconfined_u:object_r:httpd_sys_content_t:s0 /var/www/html/test.html
[root@dsbalakireva ~]# chcon -t samba_share_t /var/www/html/test.html
[root@dsbalakireva ~]# ls -Z /var/www/html/test.html
unconfined_u:object_r:samba_share_t:s0 /var/www/html/test.html
[root@dsbalakireva ~]#
```

Рис. 3.10: Смена контекста

При повторной попытке открыть файл через веб-браузер получаем ошибку доступа (рис. 3.11).

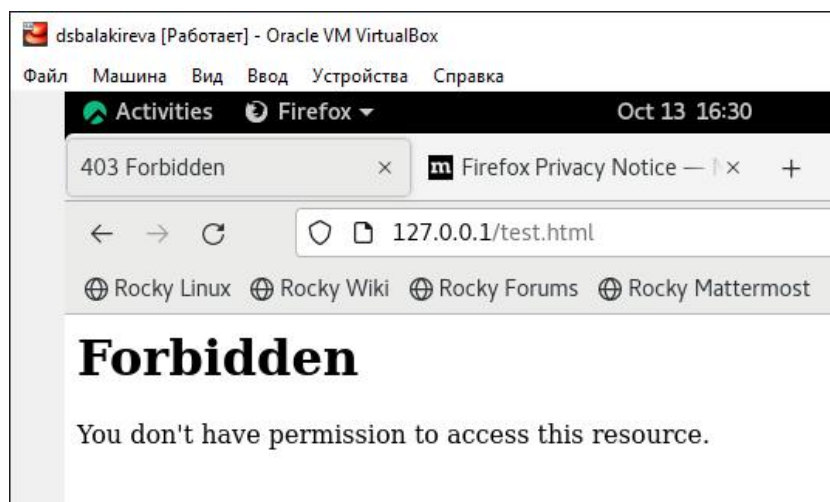


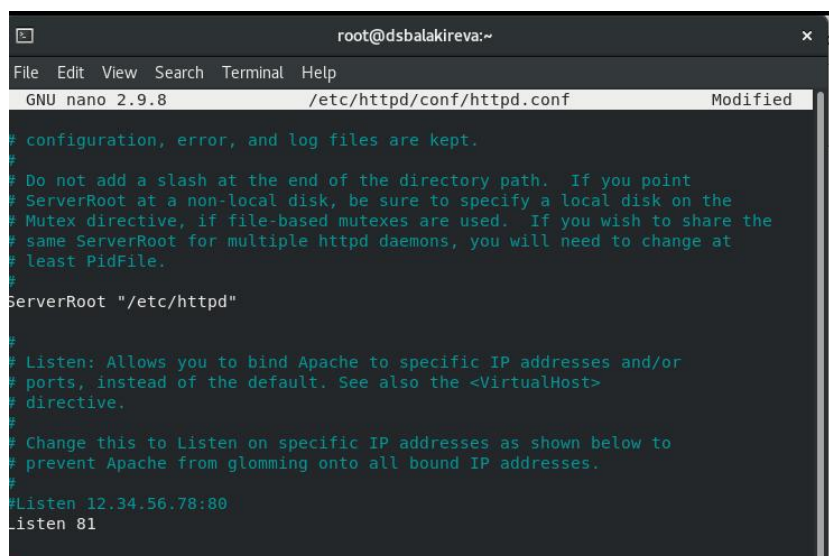
Рис. 3.11: Ошибка доступа

Убеждаемся, что файл доступен для чтения всем пользователям командой `ls -l`. Далее смотрим log-файлы веб-сервера Apache командой `tail`, где показаны ошибки (рис. 3.12).

```
[root@dsbalakireva ~]# ls -l /var/www/html/test.html
-rw-r--r--. 1 root root 34 Oct 13 16:23 /var/www/html/test.html
[root@dsbalakireva ~]# tail /var/log/messages
Oct 13 16:30:28 dsbalakireva org.gnome.Shell.desktop[1738]: libinput error: client bug: timer event3 debounce: scheduled expiry is in the past (-118ms), your system is too slow
Oct 13 16:30:28 dsbalakireva org.gnome.Shell.desktop[1738]: libinput error: client bug: timer event3 debounce short: scheduled expiry is in the past (-134ms), your system is too slow
Oct 13 16:30:28 dsbalakireva setroubleshoot[42276]: /sys/fs/selinux/policy is in use by another process. Exiting!
Oct 13 16:30:37 dsbalakireva dbus-daemon[787]: [system] Activating service name='org.fedoraproject.Setroubleshootd' requested by ':1.472' (uid=0 pid=750 comm="/usr/sbin/sedispatch" label="system_u:system_r:auditd_t:s0") (using servicehelper)
Oct 13 16:30:42 dsbalakireva dbus-daemon[787]: [system] Successfully activated service 'org.fedoraproject.Setroubleshootd'
Oct 13 16:30:50 dsbalakireva setroubleshoot[42296]: failed to retrieve rpm info for /var/www/html/test.html
Oct 13 16:30:50 dsbalakireva dbus-daemon[787]: [system] Activating service name='org.fedoraproject.SetroubleshootPrivileged' requested by ':1.477' (uid=991 pid=42296 comm="/usr/libexec/platform-python -Es /usr/sbin/setroub" label="system_u:system_r:setroubleshootd_t:s0-s0:c0.c1023") (using servicehelper)
Oct 13 16:30:54 dsbalakireva dbus-daemon[787]: [system] Successfully activated service 'org.fedoraproject.SetroubleshootPrivileged'
Oct 13 16:31:01 dsbalakireva setroubleshoot[42296]: SELinux is preventing httpd from getattr access on the file /var/www/html/test.html. For complete SELinux me
```

Рис. 3.12: Ошибки в log-файлах

Устанавливаем веб-сервер Apache на прослушивание TCP-порта 81, изменяя строку `Listen` в файле `/etc/httpd/conf/httpd.conf` (рис. 3.13).

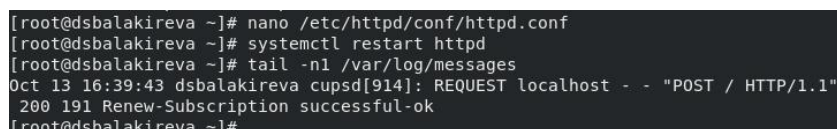
A terminal window titled 'root@dsbalakireva:~' showing the nano editor editing '/etc/httpd/conf/httpd.conf'. The file is marked as 'Modified'. The visible content includes comments about configuration, error, and log files; a warning about directory paths; the 'ServerRoot "/etc/httpd"' directive; and the 'Listen' directive being changed from '12.34.56.78:80' to '81'.

```
root@dsbalakireva:~
File Edit View Search Terminal Help
GNU nano 2.9.8 /etc/httpd/conf/httpd.conf Modified

# 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 specific IP addresses as shown below to
# prevent Apache from glomming onto all bound IP addresses.
#
#Listen 12.34.56.78:80
Listen 81
```

Рис. 3.13: Прослушивание 81 порта

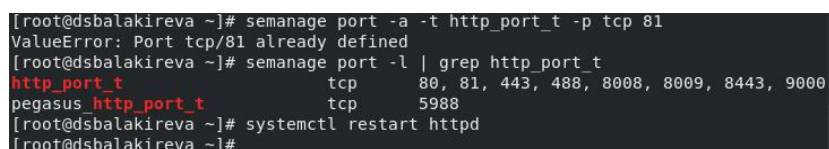
Перезапускаем сервер и смотри данные log-файлов веб-сервера Apache (рис. 3.14).

A terminal window showing a sequence of commands: 'nano /etc/httpd/conf/httpd.conf', 'systemctl restart httpd', and 'tail -n1 /var/log/messages'. The output of the tail command shows a successful POST request.

```
[root@dsbalakireva ~]# nano /etc/httpd/conf/httpd.conf
[root@dsbalakireva ~]# systemctl restart httpd
[root@dsbalakireva ~]# tail -n1 /var/log/messages
Oct 13 16:39:43 dsbalakireva cupsd[914]: REQUEST localhost - - "POST / HTTP/1.1"
200 191 Renew-Subscription successful-ok
[root@dsbalakireva ~]#
```

Рис. 3.14: Перезапуск сервера

Устанавливаем для веб-сервера Apache порт TCP-81 и проверяем его наличие в списке портов командой semanage (рис. 3.15).

A terminal window showing the command 'semanage port -a -t http_port_t -p tcp 81'. It returns an error 'ValueError: Port tcp/81 already defined'. Then 'semanage port -l | grep http_port_t' is run, showing a list of ports including 81. Finally, 'systemctl restart httpd' is executed.

```
[root@dsbalakireva ~]# semanage port -a -t http_port_t -p tcp 81
ValueError: Port tcp/81 already defined
[root@dsbalakireva ~]# 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@dsbalakireva ~]# systemctl restart httpd
[root@dsbalakireva ~]#
```

Рис. 3.15: Установка порта

Возвращаем файлу test.html контекст httpd_sys_content_t и снова успешно просматриваем страницу в веб-браузере (рис. 3.16).

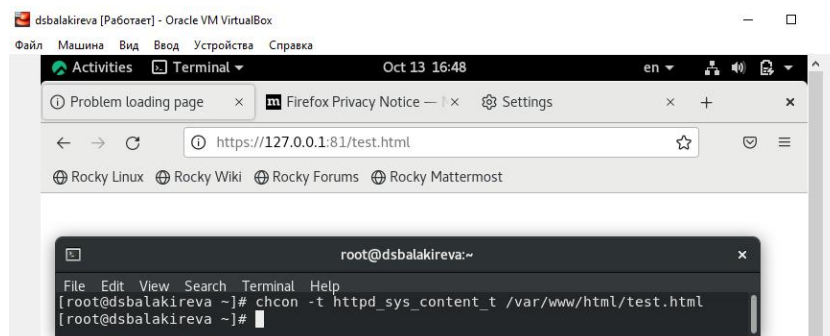


Рис. 3.16: Повторный просмотр файла в веб-браузере

Возвращаем в конфигурационный файл прослушивание порта 80 и удаляем порт 81 из списка портов (рис. 3.17).

```
[root@dsbalakireva ~]# nano /etc/httpd/conf/httpd.conf
[root@dsbalakireva ~]# semanage port -d -t http_port_t -p tcp 81
ValueError: Port tcp/81 is defined in policy, cannot be deleted
[root@dsbalakireva ~]# 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@dsbalakireva ~]# cat /etc/httpd/conf/httpd.conf | grep "Listen"
# Listen: Allows you to bind Apache to specific IP addresses and/or
# Change this to Listen on specific IP addresses as shown below to
#Listen 12.34.56.78:80
Listen 80
[root@dsbalakireva ~]#
```

Рис. 3.17: Удаление порта

Удаляем файл test.html (рис. 3.18).

```
[root@dsbalakireva ~]# rm /var/www/html/test.html
rm: remove regular file '/var/www/html/test.html'? y
[root@dsbalakireva ~]# ls /var/www/html
[root@dsbalakireva ~]#
```

Рис. 3.18: Удаление файла

4 Выводы

Я получила основные навыки администрирования в ОС Linux и проверила работу SELinux на практике совместно с веб-сервером Apache.

Список литературы

1. GNU Bash Manual [Электронный ресурс]. Free Software Foundation, 2016.
URL: <https://www.gnu.org/software/bash/manual/>.